REMARKS

Reconsideration of the above-identified patent application in view of the remarks following is respectfully requested.

Claims 1-19 are pending in the application. In the Office Action dated May 9, 2008, claims 1-19 were rejected. The rejection is respectfully traversed. Applicant hereby amends claims 1, 3, 6, 9, 10, 14, 15, 16, 17 and 19 by adding the word "doorbell" before "buffer" (claims 1, 3, 6, 10, 14, 16, 17 and 19), "unsynchronized" before "kernel-free" (claim 15) and/or "logical" before "common path" (claims 9, 10). The amendment is just for making the claim language clearer, without changing the meaning or the scope in any way.

The invention in the above-identified patent application discloses a method and system for preventing deadlock in communication between a host software application and a network interface card (NIC). The method includes writing a doorbell associated with at least one descriptor having a descriptor context to a doorbell buffer in the NIC, dropping at least one doorbell from the doorbell buffer if the doorbell buffer is full, thereby allowing a write of a new doorbell to the doorbell buffer, and recovering each dropped doorbell for further execution of descriptors associated with this doorbell. The descriptor execution is in order of posting by the application to the NIC. The invention enables the use of a single logical communication path common to doorbell writes and context read responses without a kernel call and facilitates unsynchronized, kernel-call-free, unrestricted traffic along the single common path.

The present invention lists the problems associated with commonly used protocols/systems in its Background section, referring specifically to US Patent application 20020165897 by Kagan et al. [paragraph 0009]. In one commonly used protocol, application software (SW) writes a descriptor to a buffer in system memory and writes a doorbell that prompts the NIC HW to execute this descriptor to a doorbell buffer, preferably a FIFO buffer, in the NIC. The NIC <u>reads the queue pair (QP) context from a locally attached</u> (directly attached to the NIC) <u>memory</u> and the descriptor is executed. The doorbell writes acceptance by the NIC HW is unconditional - the basic assumption is that each doorbell write is accepted as it arrives. The system makes sure that read responses needed to process this doorbell use a <u>different</u> path than writes, thus preventing

deadlock. However, this commonly used system has a main disadvantage in the need for an additional, separate memory attached to the NIC.

Prior art systems that use a single write/read path were known and were discussed in the above-identified patent application. In such a system, the software must guarantee that the doorbell FIFO buffer is never full. This guarantee is provided by synchronizing all consumers through the OS, i.e. by using a kernel call. Disadvantageously, this implies restricted access to the NIC HW, and inherent increased overhead requirements.

In contrast with the prior art, the present invention provides a solution to the deadlock as described in paragraph [0027]:

> *In order to prevent the doorbell write/context read response deadlock described above, the method, system and protocol of the present invention enable the NIC to drop some doorbells, thereby "cleaning up" a clogged read response path. This means that only one logical path without use of a kernel call is needed for both doorbell write and QP read response processes. This is a paradigm shift, representing a key novel and innovative feature of the method. We call this common path a "kernel call-free" path. All doorbells (including the dropped ones) are written by SW to (replicated in) a doorbell record, prior to ringing the doorbell. The DB record is a data structure located in system memory. This replication can be used by the NIC when it has to recover dropped (or "recovering from dropping") doorbells. Recovery occurs without neither the NIC nor the CPU having their operation affected during the recovery period (minimal intervention). Dropped doorbells are recovered without affecting other connections for which doorbells were not dropped. This is another key novel and innovative feature of the method. <u>The mechanism used for dropping and recovering doorbells is referred to hereafter as "kernel call-free" drop and recovery mechanism.</u>*

Prior to the present patent application, solutions to the DB write/QP context read response deadlock problem were based either on the use of separate write and read response paths, or on synchronization between consumers using a kernel call. There was no prior art that allowed unsynchronized (without a kernel call or "kernel-call free"), unrestricted traffic of read responses/writes <u>along a single common logical path</u>. The solution provided by this invention - user level, kernel-free access to the NIC together with a single write/read response logical path - was previously unknown.

§ 102 Rejections

Claims 1-19 were rejected under 35 U.S.C. 102(b) as being anticipated by Kagan et al. (hereinafter Kagan), US Publication No. 2002/0165897. The rejection is respectfully traversed. Applicant's previous arguments, repeated below, were rejected by the Examiner. Applicant respectfully submits that the Examiner has consistently misstated the content of the Kagan reference and the meaning of the following key features claimed in the present invention: a) "doorbell buffer" from which doorbells are dropped (claims 1 and 19), and b) a single logical path which allows unrestricted unsynchronized, kernel-free traffic (claims 9 and 15).

Kagan discloses an HCA (NIC) that provides a host processor with two complementary modes of submitting descriptors to be executed by the adapter: a normal mode, in which the host writes descriptors to a system memory and rings an assigned doorbell to notify the adapter; and a priority mode, in which the host writes the descriptor itself to a doorbell address of the adapter. In the priority mode, the adapter is relieved of the need to read the descriptor from the memory, and can thus begin execution as soon as it has resources available to do so. Because the adapter typically has limited buffer space available to hold descriptors awaiting processing, the host is preferably programmed to restrict its own use of the priority mode.

Kagan's FIG. 1 illustrates an InfiniBand (IB) network communication system having a host channel adapter (HCA or NIC) 22. The functional elements of HCA 22 are shown in his FIG. 2 and include a doorbell (FIFO) buffer 44 and a priority descriptor buffer 46. Descriptor buffer 46 is not a doorbell buffer. This is made abundantly clear in, among others, his paragraphs [0057], [0060] and [0062]. A doorbell is a signal (command) provided to the HCA to trigger the execution of a new task, while a descriptor is the description of the task (address of data to be transferred, destination, length, etc). Kagan discloses a mechanism to simultaneously submit a descriptor together with its doorbell to speed up its execution. It would be immediately clear to one skilled in the art that a descriptor buffer is essentially different from a doorbell buffer, and that a doorbell buffer is never used to store descriptors and vice versa. In Kagan, while the priority (descriptor) buffer may be emptied of its contents (therefore "dropping descriptors", not doorbells), doorbell buffer 44 never drops any doorbell. The "dropping"

of doorbells is irrelevant to Kagan's invention and therefore never mentioned or discussed.

The doorbell dropping mechanism of the present invention facilitates unrestricted, unsynchronized, kernel-free traffic along a single logical path, as stated in paragraph [0031]:

> *Doorbells received by the NIC HW from the host SW are temporally stored in a doorbell buffer, preferably a FIFO buffer 428. As a result, read responses involve traffic 430 between the system memory and the NIC and therefore share a single logical path 432 through bus 414 with writes 434 from CPU 404. Without the provision of the deadlock resolution method disclosed herein, such a system must have user synchronization employing a kernel call to prevent deadlock, as explained above.*

In Kagan, context information required to process doorbells is read from a locally attached memory (not from system memory 32) and therefore it comes through a separate logical (and physical) path than writes. His paragraph 0052 mentions the locally attached memory:

> *"Preferably, HCA 22 comprises a single-chip device, including one or more embedded microprocessors and memory on-board".*

Also the end of his paragraph 0054 touches on the context information read where it says:

> *"In response to the doorbell, the HCA retrieves context information regarding the QP, and then and then reads and executes the descriptors"*

Thus, Kagan neither discloses a doorbell buffer that drops doorbells nor unsynchronized, kernel-free traffic of doorbell writes and read responses on a single logical path. Since Kagan fails to disclose key limitations recited in the independent claims of the present invention, he cannot and does not anticipate the present invention. Moreover, Kagan does not even render the present invention obvious.

Returning now to the specific rejections:

In reference to claim 1, Examiner states that Kagan teaches a method/system comprising the step of "if said buffer is full, dropping at least one doorbell from said buffer, thereby allowing a write of a new doorbell to said buffer", referring to paragraph 13 in Kagan. Paragraph 13 reads:

> *In preferred embodiments of the present invention, a network interface adapter provides a host processor with two complementary modes of submitting descriptors to be executed by the adapter: a normal mode, in which the host writes descriptors to a system memory and rings an*

assigned doorbell to notify the adapter; and a priority mode, in which the host writes <u>the descriptor</u> itself to a doorbell address of the adapter. In the priority mode, the adapter is relieved of the need to read the descriptor from the memory, and can thus begin execution as soon as it has resources available to do so. Because the adapter typically has limited buffer space available to hold descriptors awaiting processing, the host is preferably programmed to restrict its own use of the priority mode. Most preferably, when the host attempts to write a descriptor to the adapter doorbell in priority mode, it also writes the descriptor to the system memory, so that the adapter can execute the descriptor in the normal mode when it is not able to carry out the requested priority processing.

As indicated above, Applicant respectfully submits that Kagan does not teach dropping doorbells from a <u>doorbell buffer</u> at all, but teaches <u>dropping descriptors from a descriptor buffer</u>. Paragraph 13 in Kagan deals with descriptor buffer 46, not doorbell buffer 44. There is absolutely no mention or indication anywhere in Kagan of a <u>doorbell</u> being dropped if a <u>doorbell buffer</u> is full. Therefore, Kagan cannot and does not anticipate the invention claimed in claim 1. Moreover, Kagan cannot and does not even render the invention claimed in claim 1 unpatentable.

Claim 2-8 depend directly or indirectly from claim 1 and include all of its limitations. Therefore, Kagan cannot and does not anticipate the invention claimed in claims 2-8, because each of these claims includes a limitation not disclosed by Kagan.

In reference to claim 9, Examiner states that Kagan teaches a method/system including providing a single logical communication path common to doorbell writes and context read responses [paragraph 7] and facilitating unsynchronized, kernel call-free unrestricted traffic along said single common path. Paragraph 7 reads:

A host connects to the IB fabric via a network adapter, which is referred to in IB parlance as a host channel adapter (HCA). When an IB "consumer," such as an application process on the host, needs to open communications with some other entity via the IB fabric, it asks the HCA to provide the necessary transport service resources by allocating a transport service instance, or queue pair (QP), for its use. Each QP has a send queue and a receive queue and is configured with a context that includes information such as the destination address (referred to as the local identifier, or LID) for the QP, service type, and negotiated operating limits. Communication over the fabric takes place between a source QP and a destination QP, so that the QP serves as a sort of virtual communication port for the consumer.

Applicant submits that paragraph 7 in Kagan does not relate to Kagan's invention, but to the general state of InfiniBand, being part of the Background section. Paragraph 7 does not mention context read responses at all. While Kagan supports unsynchronized,

kernel call-free traffic, this traffic is along multiple paths, not along a single common logical path. There is nothing in paragraph 7 or elsewhere in Kagan that refers to an unsynchronized, kernel call-free unrestricted traffic along a single common logical path. Therefore Kagan cannot and does not anticipate the invention claimed in claim 9. Moreover, Kagan cannot and does not even render the invention claimed in claim 9 unpatentable. Similarly, Kagan cannot and does not anticipate the invention claimed in any of claims 10-14, which depend directly or indirectly from claim 9 and include all of its limitations.

In reference to claim 15, the Examiner states that Kagan teaches a system comprising all three elements recited in the claim, citing paragraphs 7 and 52 for element (a) and paragraph 52 for elements (b) and (c). Paragraph 52 reads:

> *FIG. 1 is a block diagram that schematically illustrates an InfiniBand (IB) network communication system 20, in accordance with a preferred embodiment of the present invention. In system 20, a host channel adapter (HCA) 22 couples a host processor 24 to an IB network (or fabric) 26. Preferably, HCA 22 comprises a single-chip device, including one or more embedded microprocessors and memory on-board. Alternatively, multi-chip implementations may be used. Typically, host 24 comprises an Intel Pentium™ processor or other general-purpose computing device with suitable software. Host 24 interacts with HCA 22 by opening and manipulating queue pairs (QPs), as provided by the above-mentioned IB specification. HCA 22 typically communicates via network 26 with other HCAs, as well as with target channel adapters (TCAs) connected to peripheral devices (not shown in the figures).*

Applicant submits that in this paragraph or in paragraph 7, Kagan does not mention context read responses or a single logical path at all. *Mutatis mutandis*, he does not mention a kernel call-free mechanism for facilitating unsynchronized, kernel-free traffic along said single logical path. Therefore Kagan cannot and does not anticipate the invention claimed in claim 15. Moreover, Kagan cannot and does not even render the invention claimed in claim 15 unpatentable. Similarly, Kagan cannot and does not anticipate the invention claimed in any of claims 16-18, which depend directly from claim 15 and include all of its limitations.

In reference to claim 19, the Examiner states that Kagan teaches a method/system comprising: responsive to a first check, dropping at least one doorbell from the doorbell buffer if the buffer is full, thereby providing space in the buffer for a respective at least one new doorbell [paragraph 13, lines 11-18]; recovering each dropped doorbell and

executing its respective associated descriptors [paragraph 13]; and responsive to same said first check, if said doorbell buffer is not full, checking if a doorbell is a repeat doorbell, and executing descriptors of each doorbell found to be not a repeat doorbell [paragraphs 14 and 54]. Applicant reiterates his arguments re. claim 1: paragraph 13 in Kagan deals with a descriptor buffer, not a doorbell buffer. There is absolutely no mention or indication any where in Kagan of a doorbell being dropped if a doorbell buffer is full. Therefore, Kagan cannot and does not anticipate the invention claimed in claim 19. Moreover, Kagan cannot and does not even render the invention claimed in claim 19 unpatentable.

In view of the above amendments and remarks it is respectfully submitted that Claims 1-19 are now in condition for allowance. Prompt notice of allowance is respectfully and earnestly solicited.

Respectfully Submitted

Mark M. Friedman
Attorney for Applicant
Registration No. 33,883

Date: November 6, 2008